applied sciences

MDPI

Article

A Linear Programming Method for Finding a Minimal Set of Axial Lines Representing an Entire Geometry of Building and Urban Layout

Sung Kwon Jung [1] **and Youngchul Kim** [2,*]

[1] CS Enertech, Seoul 06730, Korea; jskstrm@umich.edu
[2] KAIST Urban Design Lab, KAIST Smart City Research Center, Department of Civil and Environmental Engineering, KAIST, Daejeon 34141, Korea
* Correspondence: youngchulkim@kaist.ac.kr; Tel.: +82-42-350-3636

Received: 19 May 2020; Accepted: 8 June 2020; Published: 22 June 2020

Abstract: This paper devises an algorithm for finding the minimal set of axial lines that can represent a geometry of building and urban layout in two dimensions. Although axial lines are useful to analyze spatial configuration in the Space Syntax, existing methods for selecting axial lines seldom address the optimality of their solutions. The proposed algorithm uses linear programming to obtain a minimal set of axial lines. To minimize the number of axial lines that represent the entire geometry of building and urban layout, a linear programming problem is established in which a set of axial lines represents the entire geometry. The axial lines must have at least one intersection with every extension line of the wall edges to the sides of the reflex angles. If a solution to this linear programming problem exists, it will be guaranteed to be an optimum. However, some solutions of this general linear programming problem may include isolated lines, which are undesirable for an axial line analysis. To avoid isolated axial lines, this paper states a new formulation by adding a group of constraints to the original formulation. By examining the modified linear programming problem in various two-dimensional building maps and spatial layouts, this paper demonstrates that the proposed algorithm can guarantee a minimum set of axial lines to represent a two-dimensional geometry. This modified linear programming problem prevents isolated axial lines in the process of axial line reduction.

Keywords: space syntax; axial line; spatial analysis; modeling; optimization; computational modeling

1. Introduction

An axial line is a geometrical element that represents a convex space, i.e., a spatial unit comprising the entire geometry of building and/or urban layout. An intersection between two axial lines thus indicates the link of the convex spaces represented by the axial lines. By using an axial map of a built environment, people can recognize the abstract spatial configuration of linear two-dimensional shapes [1,2]. An axial line is defined as the longest line that can be drawn in a divided convex space that links to other axial lines that cover all of the given space without repetition [1]. An axial line is also considered to be a linear representation of a spatial configuration [2]. Thus, an axial map with the minimal set of axial lines characterizes a spatial, geometric and topological relationship [3]. As axial lines link the convex spaces within building interiors and urban layouts, these lines connect with other axial lines to show the spatial relationship of connectivity between convex spaces. These connections of axial lines as an axial map describe the geometries of a building or an urban space. Thus, this paper is based on the concept that a minimal set of axial lines is an axial map and that these axial lines represent the topological connectivity of geometric divisions in the built environment. However, many studies still manually create axial lines or choose generating all axial lines that hardly illustrate representative spatial connections. Conventional methods to select axial lines among all axial lines were based on greedy elimination algorithm and whether or not such selections are minimal is unclear [2].

This paper develops an algorithm for finding a minimal set of axial lines that can represent a geometry of building and urban layout in two dimensions. Peponis et al. [2] stated that they had no proof that the axial lines selected by their greedy algorithm actually represented the minimum set of axial lines. As these researchers could not readily solve the set cover problem algorithmically, they used an approximated greedy algorithm. This method, however, could not promise to identify the real minimum of axial lines needed to represent an entire geometry of building and urban layout. If the minimum set of axial lines can be found, then an axial line map with the fewest and longest lines can be achieved that correctly guarantees the minimum set of axial lines in a building. Although Peponis et al. [2] noted the computational difficulty of the set cover problem, this paper seeks to overcome this difficulty and fully represent the geometry of built spaces.

To represent a geometry of building and urban layout by axial lines, this paper requires a linear programming method for finding a global optimum and minimal set of axial lines. As a set of axial lines describes a spatial configuration of two-dimensional shapes, such sets of lines have been used for analyzing the spatial characteristics of buildings and urban places. Many studies still use manually drawn axial lines for spatial analyses and therefore many researchers have investigated methods for generating axial lines with computer programs (e.g., [2–6]). Computer programs provide the advantage of consistent results, high speed, large capacity and error immunity. In addition, some studies have adopted the approach of using all-line axial maps, because these maps are easy to generate with computer programs (e.g., [7,8]). However, although an all-line axial map is easy to create, the redundancy of axial lines in these maps often renders direct use of the maps impractical or counter-intuitive. Thus, due to the strong demand for a scientific approach, it is important to select a set of axial lines from an all-line axial map.

2. Background

2.1. *Axial Line Analysis Computation*

To develop a method of drawing axial lines, several previous studies have suggested the use of all-line axial maps to show spatial configurations (e.g., [2,3]). All-line axial maps are drawn from a set of vertex-to-vertex lines within a given space that do not intersect the wall lines or boundaries of that space [2,3]. The rules for these maps are intuitive and allow the maps to be easily generated by computer programs. Thus, by following these mapping rules, this paper develops a method of axial line reductions for finding the minimal set of axial lines.

There are, however, other approaches to generating axial maps (e.g., [4–6]). Batty and Rana [4] and Jiang and Liu [5] used visual fields, isovists or polygons in given spaces to generate axial lines. These approaches relied on spatial division by polygons and focused on generating axial lines based on visibility, rather than reducing axial lines from the vertex-to-vertex lines of an all-line axial map. Liu and Jiang [6] generated axial lines by using existing center lines in urban maps and layouts based on open-source maps. Although these approaches have proven useful for generating axial lines, they developed a new application unlike the reduction methods from all-line axial map that are used in well-known space syntax programs. Rather than developing a new application, this paper adapts an existing method for generating maps of all axial lines. Since this paper investigates a method for finding a minimal set of axial lines from all axial lines, Batty and Rana's and Jiang and Liu's approaches are therefore outside the scope of this paper. Thus, this paper uses the method of generating all-line axial maps and reducing the axial lines of these maps to find their minimal sets of lines. This paper seeks to add an optimization perspective, which focuses on finding a fewest set, to other approaches for generating axial lines including those proposed by Batty and Rana [4], Jiang and Liu [5] and Liu and Jiang [6].

Many previous studies on reducing axial lines have used a greedy backward elimination and a sub-set elimination. Peponis et al. [2] and Turner et al. [3] demonstrated how to reduce axial lines in all-line maps for axial analysis. They used a greedy backward elimination and a sub-set elimination to reduce an all-line axial map to a simpler set of axial lines. The greedy and subset algorithms were required to examine every possible combination of lines. However, because these methods eliminated axial lines in a stepwise manner, they did not always result in a minimal set of axial lines. The results of these algorithms significantly depended on the manner of initiating the combinations for each set of axial lines and the manner of setting up the next procedure to select another axial line. Therefore, this paper seeks to find a holistic approach to identify a minimal set of axial lines.

2.2. Linear Programming

A linear programming is an algorithm for solving an optimization problem whose objective and constraints involve linear relationships. The canonical form of the linear programming problem without equality constraints is

maximize $\mathbf{CX}$
subject to
$\mathbf{AX} \leq \mathbf{B}$

Formulation 1. Linear programming problem in a canonical form
($\mathbf{A}$: matrix; $\mathbf{B}$, $\mathbf{X}$: column vectors; $\mathbf{C}$: row vector)

Depending on the variables used, this problem can be categorized into various linear programming problems. If some of the variables involved are integers, then it is a mixed-integer linear programming problem. If all of the variables involved are integers, then it is an integer-linear programming problem. If every variable involved is binary (zero or one), then it is a zero-one linear programming problem.

Linear programming has been used in urban planning, mostly for space/land allocation [9–12], site planning [13] and project scheduling [14]. By solving linear programming problems, previous studies have demonstrated automatic selection and an efficient approach to using computer programs for calculating large data. Although few studies have used linear programming for spatial analysis, this process can also be defined as a linear programming problem for determining the axial lines in an all-line axial map. As all of the variables in axial line reduction problems are binary (they determine whether or not an axial line is in the reduced set), this paper seeks to find a solution to the problem by using the zero-one integer linear programming method.

Therefore, this paper finds a minimal set of axial lines representing an entire geometry of building and urban layout by use of a linear programming method, rather than by a stepwise elimination. In this paper, linear programming is the main method for obtaining a minimal set of axial lines. The linear programming problem is formulated to minimize the number of axial lines that represent an entire geometry of building and urban layout.

3. Methods

The overall aim of this paper is to develop a novel method for generating an axial map to represent an entire geometry of building and urban layout. By using the proposed method, a minimum set of axial lines will be selected from an all-line axial map. This map of selected axial lines will be guaranteed to represent the minimum set of axial lines for a given building's interior spaces. The objectives are to devise a linear programming method for finding a minimal set of axial lines from all-line axial maps, to formulate a modified linear programming problem to solve the issues encountered when adopting a linear programming method, to deliver a modified linear programming method that demonstrates a holistic approach to minimizing axial lines, and to deliver a computational tool with an algorithm that automatically identifies a set of axial lines in building spaces.

This paper starts with problem formulation for defining spatial compositions that illustrate the spatial components of a building's interior space and then finds a minimum set of axial lines.

Based on the problem formulation, the research follows a problem-solving procedure, a modified problem-solving procedure, and computation. For each step, the proposed algorithms are tested on the collected cases that have been used in Peponis et al. [2]'s study of axial line maps.

4. Problem Formulation

A minimum set of axial lines can be found when every axial line has at least one intersection with the extension lines from the wall edges to the sides of the reflex angles among n axial lines and m extension lines. This assumption follows a concept proposed by Peponis et al., that such a set of axial lines represents the entire building's geometry by including all spaces and "getting everywhere" in the building [2]. According to the definitions used by Peponis et al. [2], s-lines (or "surface lines") are extension lines of the wall edges to the sides of reflex angles, and m-lines (or "movement lines") are diagonals of straight lines between the vertexes, excluding lines that intersect walls in a given space. Peponis et al. proposed that to include all of the geometry (or "get everywhere"), it is necessary that the m-lines cross every s-line, and that this set of m-lines becomes a set of axial lines that represent a building's geometry [2]. If every s-line has at least one intersection with n axial lines and m s-lines, then a set of axial lines can represent an entire building's geometry by including all spaces (or getting everywhere). This circumstance indicates that if a solution to the linear programming problem exists, then it will be guaranteed to be an optimum solution. The minimal set of axial lines calculated using this formulation will have intersections with every s-line. This special form of linear programming, namely zero-one integer linear programming, is adopted in this paper.

This paper's method assumes that there are n axial lines and m s-lines in a given building plan. Let $x_i = 1$ when the ith axial line is selected (or placed in a reduced set), and let $x_i = 0$ otherwise, because the number of axial lines is to be minimized and the objective function of the problem is to minimize the sum of n variables (min $\sum_{i=1}^{n} x_i$). For a formulation of the optimization problem, this paper uses the formation of intersection with a set of n axial lines. Intersection with a set of n axial lines is equivalent to intersection with at least one axial line in the set. To check whether an s-line has such an intersection, Equation (1) is used.

$$a_1 + a_2 + \cdots + a_{n-1} + a_n \geq 1 \tag{1}$$

where $a_i = 1$ if an s-line has an intersection with the ith of n axial lines, and 0 otherwise.

In addition, the equation for an intersection with a subset is as follows:

$$a_1x_1 + a_2x_2 + \cdots + a_{n-1}x_{n-1} + a_nx_n \ \geq 0 \tag{2}$$

where $x_i = 1$ if an element is of the subset, and 0 otherwise.

Suppose that an s-line has intersections with a subset of axial lines. That Equation (2) is greater than or equal to 1 is the necessary and sufficient condition for this s-line to intersect at least one axial line in the reduced subset. These equations indicate a way to formulate the optimization problem in this paper. Accordingly, Formulation 2 of the optimization problem can be stated as follows:

minimize $\sum_{i=1}^{n} x_i$,

subject to

$\mathbf{AX} \geq \mathbf{B}$,

where

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n-1} & a_{1,n} \\ a_{2,1} & a_{2,2} & & & a_{2,n} \\ \vdots & & \ddots & & \vdots \\ a_{m-1,1} & & & a_{m-1,n-1} & a_{m-1,n} \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n-1} & a_{m,n} \end{bmatrix}, \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}, \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{m-1} \\ b_m \end{bmatrix},$$

$a_{ij} = 1$ if the ith s-line has an intersection with the jth axial line, and 0 otherwise,

$b_i = 1$,

$x_i \in \{0, 1\}$.

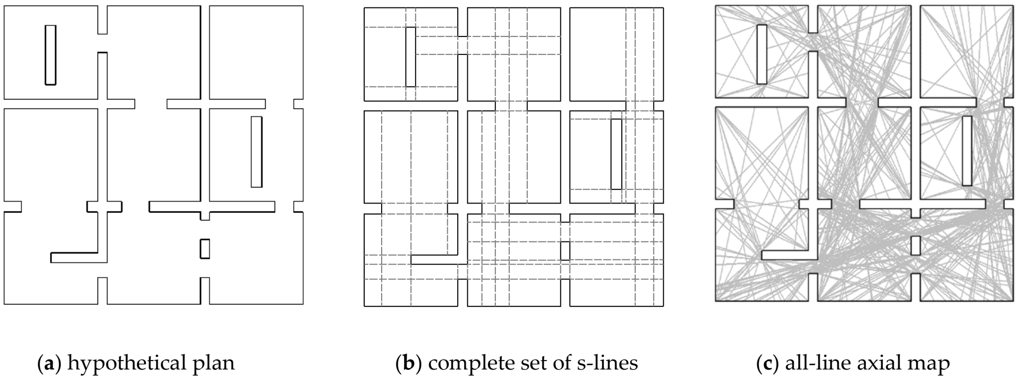
Formulation 2. Linear programming problem to find a minimal set of axial lines

This approach involves a special form of linear programming, namely zero-one integer linear programming. If a solution to this problem exists, then the solution will be guaranteed to be an optimum solution. Based on the principles of Equations (1) and (2), Formulation 2 seeks a minimal subset of axial lines, at least one of which has an intersection with each s-line. This paper investigates the problem of zero-one integer linear programming to find a minimal set of axial lines in a given spatial design.

5. Implementation

5.1. Problem-Solution Procedure

Problem-solution procedure leads to devising a linear programming method to find a minimal set of axial lines from an all-line axial map. For the problem-solution procedure, zero-one integer linear programming is examined in various plans, e.g., Figure 1a. Following the spatial configuration in the plan, an all-line axial map and all s-lines are calculated using a set of polygons that represent the building in question. For example, Figure 1b includes a complete set of s-lines (shown as gray dashed lines) and Figure 1c includes an all-line axial map (with gray lines) from the hypothetical plan. This hypothetical plan is used to demonstrate the implementation of Formulation 2, which yields a minimal set of axial lines.



(**a**) hypothetical plan (**b**) complete set of s-lines (**c**) all-line axial map

Figure 1. Hypothetical plan. (**a**) Hypothetical floor plan; (**b**) Description; (**c**) All-line axial map.

From a complete set of s-lines and an all-line axial map, as shown in Figure 1b,c, the matrix **A** described in Formulation 2 can be determined. A linear programming solver is used to determine a solution to the optimization problem described in Formulation 2 (minimize $\sum_{i=1}^{n} x_i$ subject to $\mathbf{AX} \geq \mathbf{B}$). For example, in Figure 2, the result from Formulation 2 shows that an example of a minimal set of axial lines is acquired by using a linear programming solver. In Figure 2, there are 11 axial lines in the axial map. However, an isolated axial line is found that connects to no other lines. The line in the circle of Figure 2 does not connect to other lines. This occurs because the constraints of the optimization problem do not prevent isolated axial lines. Isolated lines are undesirable for axial line analysis because every axial line in an axial map needs to be connected to others, unless its corresponding space is physically separated. Accordingly, the line in Figure 2 that is detached from the other lines needs to be connected to other lines in the circle of Figure 2. This isolated line does not fully represent the spatial geometry as an axial line. For this reason, the next procedure seeks to develop the additional constraints that need to be included in the optimization problem to prevent a solution from having undesirable isolated lines.

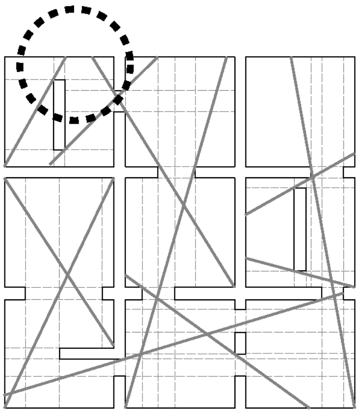


Figure 2. A minimal set of axial lines that "get everywhere" (n = 11).

5.2. Modified Problem-Solution Procedure

The modified problem-solution procedure leads to the formulation of a modified linear programming problem to resolve issues that arise when adopting a linear programming method. In using the optimization problem in Formulation 2, isolated lines can still occur in some cases, and such lines are undesirable for axial line analysis. The zero-one integer linear programming method is modified to accommodate this situation. While the original constraints of the optimal problem do not prevent isolated lines that are undesirable for axial line analysis, the isolated lines can be suppressed by adding more constraints to the optimization problem. To avoid isolated axial lines, a group of constraints are added to the original formulation. If this modified formulation of zero-one integer linear programming can provide a solution to the optimization problem, then the solution will be guaranteed to be an optimum. Therefore, an optimization problem that includes constraints for suppressing isolated lines is formulated. The new formulation of the problem (Formulation 3) is stated as follows:

minimize $\sum_{i=1}^{n} x_i$,

subject to

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix} \mathbf{X} \geq \begin{bmatrix} \mathbf{B} \\ \mathbf{D} \end{bmatrix},$$

where

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n-1} & a_{1,n} \\ a_{2,1} & a_{2,2} & & & a_{2,n} \\ \vdots & & \ddots & & \vdots \\ a_{m-1,1} & & & a_{m-1,n-1} & a_{m-1,n} \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n-1} & a_{m,n} \end{bmatrix}, \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}, \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{m-1} \\ b_m \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n-1} & c_{1,n} \\ c_{2,1} & c_{2,2} & & & c_{2,n} \\ \vdots & & \ddots & & \vdots \\ c_{n-1,1} & & & c_{n-1,n-1} & c_{n-1,n} \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n-1} & c_{n,n} \end{bmatrix}, \mathbf{D} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix},$$

$a_{ij} = 1$ if the ith s-line has an intersection with the jth axial line, and 0 otherwise,
$b_i = 1$,
$c_{ij} = 1$ if the ith axial line has an intersection with the jth axial line, and 0 otherwise,
$c_{ii} = -1$,
$d_i = 0$,
$x_i = 1$ if the ith axial line is selected, and 0 otherwise,
m = the number of s-lines,
n = the number of axial lines.

Formulation 3. Modified problem for finding a minimal set of axial lines

This modified linear programming problem is examined in a hypothetical plan (Figure 1a) and it prevents isolated axial lines in the axial line reduction like Figure 3. In the circle appearing in Figure 3, all of the lines are connected to other lines, but in Figure 2, the isolated line is not connected to other lines. Adding more constraints into the linear programming problem enables Formulation 3 to find a minimal set of axial lines without an isolated line. In addition, the number of axial lines according to the modified formulation is expected to equal to or greater than that calculated by the original formulation (which could include isolated lines). For example, as shown in Figure 3, the modified optimization problem prevents isolated axial lines and the number of lines is expected to remain the same as previously (n = 11). This output is tested for several cases discussed in other literature, e.g., [1,6]. Three cases of buildings and two cases of urban complexes are examined with this modified linear programming problem, as illustrated in Figures 4 and 5. These cases have been used to verify whether various methods in previous studies generate appropriate results in built environments [1,2].

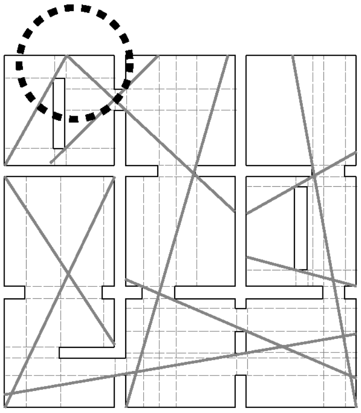


Figure 3. A minimal set of axial lines that "get everywhere" (n = 11, isolated lines suppressed).

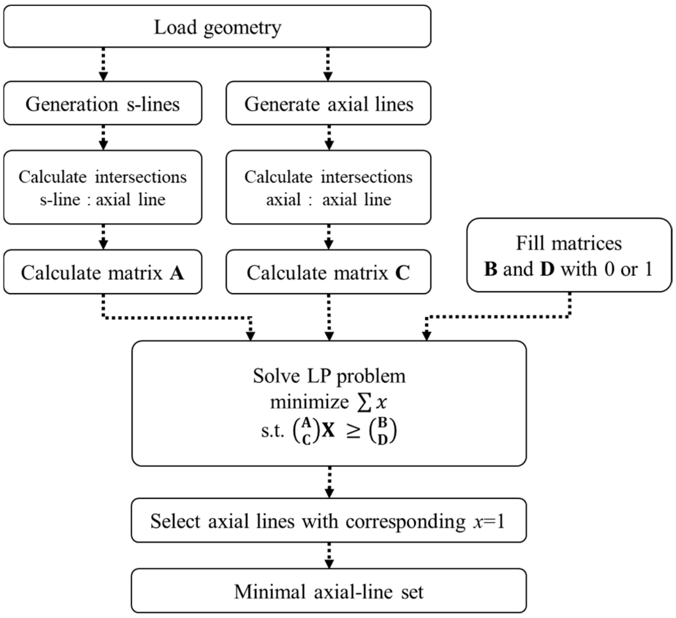
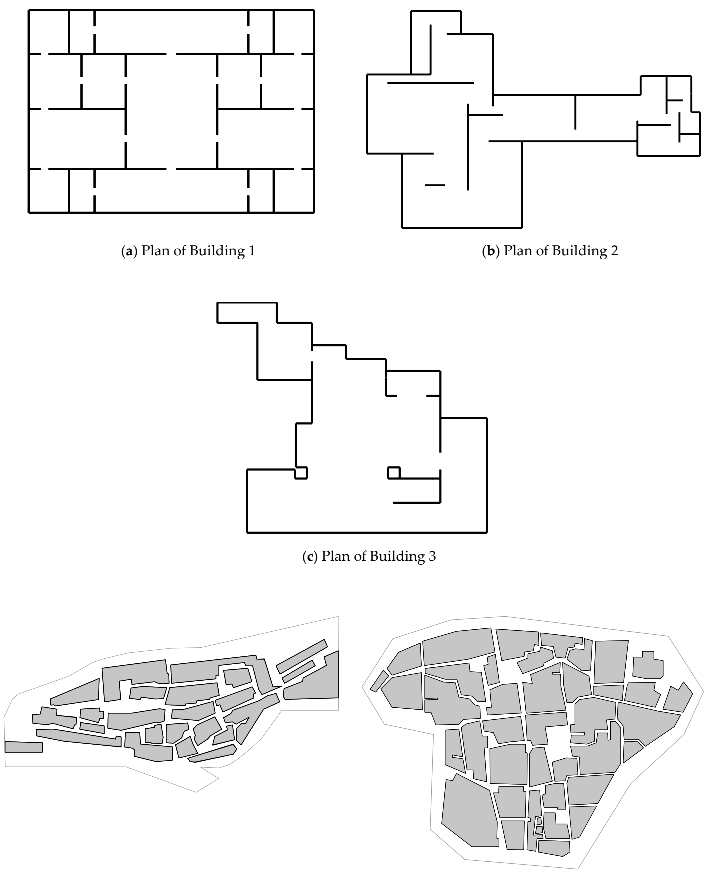

Load geometry
Generation s-lines
Generate axial lines
Calculate intersections s-line : axial line
Calculate intersections axial : axial line
Fill matrices B and D with 0 or 1
Calculate matrix A
Calculate matrix C
Solve LP problem minimize $\sum x$ s.t. $\binom{A}{C}\mathbf{X} \geq \binom{B}{D}$
Select axial lines with corresponding x=1
Minimal axial-line set

Figure 4. Computation procedure of the modified problem.



(**a**) Plan of Building 1

(**b**) Plan of Building 2

(**c**) Plan of Building 3

(**d**) Plan of Gassin

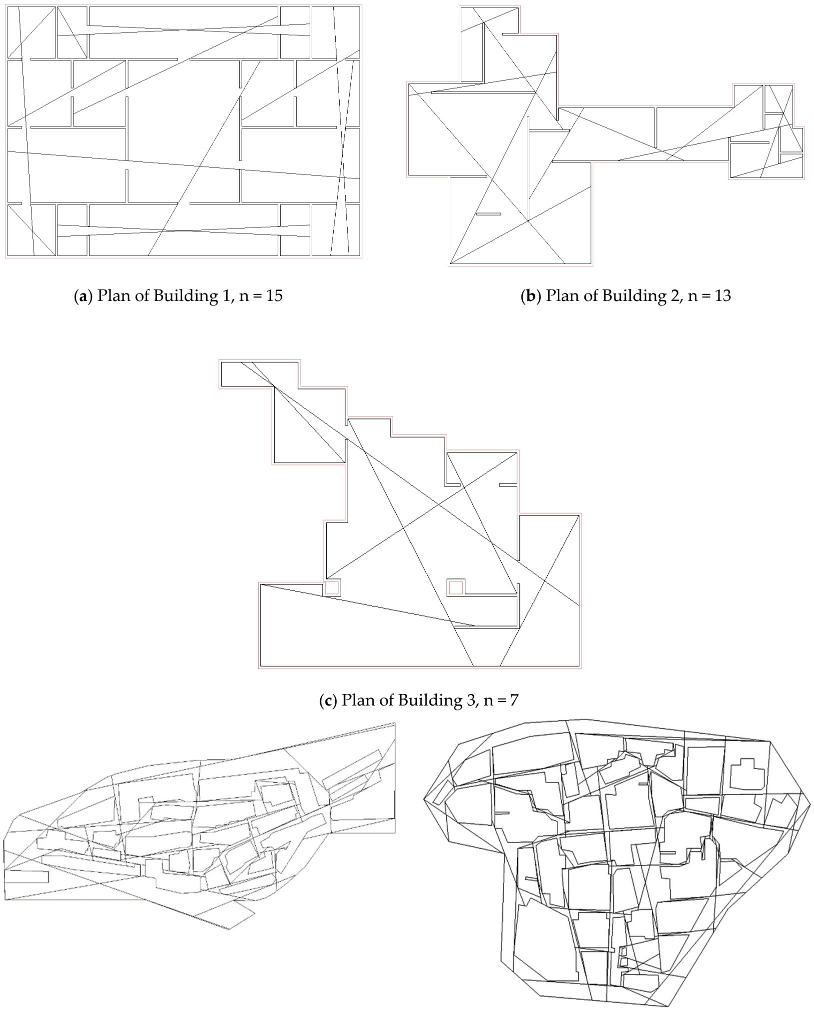
(**e**) Plan of Apt

Figure 5. Cases of Buildings and Urban Layout. (**a**) Plan of Building 1 (Source [2]) (**b**) Plan of Building 2 (Source [2]) (**c**) Plan of Building 3 (Source [2]) (**d**) Plan of Gassin (Source [2]) (**e**) Plan of Apt (Source [2]). (The authors, S.K.J and Y.K., redraw original figures in [2] with permission of the authors, Peponis et al. [2]) and the publisher, SAGE publications).

6. Computation

Computation leads to the delivery of a computational tool based on the modified linear programming method for finding a minimum set of axial lines. The problem-solution procedure follows several steps. To begin with, the s-lines and all-line axial maps are generated for a given plan. This paper uses the cases shown in Figures 4 and 5. For each combination of an axial line and an s-line, whether they have a crossing point is checked. The checking results then are used to populate matrix **A** in Formulations 3. Next, each pair of axial lines is checked to determine whether they have a crossing point, and these checking results populate matrix **C** in Formulation 3. The column vectors **B** and **D** in Formulations 3 are assigned values of 1 and then the formulations solve the given linear programming problem. The axial lines whose corresponding variables have values of 0 are eliminated. A minimal set of axial lines is generated from this procedure to get everywhere in a building.

For the computations in this paper, Syntax2D was used to calculate an all-line axial map, s-lines, and intersections among them. Syntax2D is a computer program for Space Syntax analysis that was developed at the University of Michigan, which is available at the sourceforge website (https://sourceforge.net/projects/syntax2dnew) [14]. To solve the linear programming problem in this paper, a free linear programming solver library, lp_solve 5.5, that is available at the sourceforge website (http://lpsolve.sourceforge.net/5.5/) is integrated into Syntax2D. The solutions are calculated with all pre-solve options of lp_solve active. Figure 6 illustrates minimal sets and numbers of axial lines in the collected buildings and urban layouts.



(**a**) Plan of Building 1, n = 15

(**b**) Plan of Building 2, n = 13

(**c**) Plan of Building 3, n = 7

(**d**) Plan of Gassin, n = 38

(**e**) Plan of Apt, n = 63

Figure 6. Minimal sets and numbers of axial lines in building and urban layouts. (**a**) Plan of Building 1, n = 15 (**b**) Plan of Building 2, n = 13 (**c**) Plan of Building 3, n = 7 (**d**) Plan of Gassin, n = 38 (**e**) Plan of Apt, n = 63.

It should be pointed out that the solution time may greatly vary depending on the complexity of problem, which affects the internal business of the solver library. Table 1, summarizing the computation time of the sample cases describes such an effect: the number of axial lines of Apt is similar to that of Gassin, but its solution time is far longer. A 3.6 GHz single core of a multi-core CPU was used for the computations.

Table 1. Computation time of the sample cases.

Cases	Number of Axial Lines	Data Preparation and Problem Formulation (ms)	Solution (ms)
Building 1	692	110	30
Building 2	184	10	<1
Building 3	225	10	10
Gassin	1124	532	1722
Apt	1296	740	124,389

7. Conclusions

The modified linear programming problem discussed in Section 5.2 shows a valid algorithm for finding a minimal set of axial lines that represent a given geometry. This paper devises an algorithm for finding a minimal set of axial lines that represent a given building plan. A linear programming problem was formulated based on the algorithm, then a linear programming software package was used to solve this problem. In this paper, the algorithm that uses a linear programming solver provides promising results for developing an axial line set for the spatial analysis of built environments. In addition, according to the examination of the linear programming problem in this paper, it is not necessary to use s-lines to find a minimal set of axial lines. This paper's method can use user-supplied lines or any other type of lines (believed to be) representing the entire space of interest. Given a set of lines of any type that represents the space, we can find a minimal subset of the given set representing the same space by using the linear programming method.

The most significant contribution is the provision of a novel method for automatically generating axial maps of space syntax to enable the spatial analysis and evaluation of building spaces. The developed method will have the advantage of consistent results, fast speed, large capacity and error immunity in analyzing large sets of data. Thus, this method will allow people to effectively generate axial line maps for analyzing built environments. Integrating the algorithm presented in this study into other software tools for spatial analysis is highly feasible; many linear programming solvers, free or commercial, are available. The second contribution of this paper is that it contributes to improving the quality of built environments by enabling the evaluation and analysis of spatial quality in existing or proposed construction projects. Scholars and professionals will be able to use the developed method to appropriately examine various spatial aspects in the given circumstances of built environments. With such an assessment of spatial qualities, appropriate alternatives can be proposed.

However, it is desired that every inner wall be enclosed by axial lines in order to represent the topology of the spatial configuration. While Peponis et al. [2] demonstrated that a circulation loop connection occurred in spatial configuration, the formulations in this paper leave this circulation loop connection requirement unconsidered. Therefore, some inner walls can be unenclosed by a minimal set of axial lines. Finding a set of axial lines enclosing every inner wall can follow the second step of Peponis et al. [2] once a minimal set is found by using the proposed methods in this paper. This proposed method guarantees a minimal set of axial lines representing an entire geometry of building and urban layout. For the same reason, preventing isolated groups of axial lines is unconsidered. In future work, dynamic linear programming will be sought to keep axial lines connected and incorporate "inner wall enclosure" constraints.

This paper describes the initial attempt to find a minimal set of axial lines using the optimization problem. Unlike a greedy backward elimination and a sub-set elimination, the optimization problem simultaneously examines the entire space of a given plan. This paper achieves the first step in finding a minimal set of axial lines, each of which has at least one connection to the others. Developing the approach using the linear programming method, the authors are planning to formulate a more devised problem for solving the challenges in this paper. In future research, while non-dynamic linear programming solvers are incapable of solving the circulation loop problem, it is possible that a branch-and-bound algorithm, which is one of the methods for solving an integer linear programming problem, will be devised to solve the problem. The branch-and-bound algorithm and dynamic linear programming to be devised may include a few graph search algorithms, including cycle detection and graph connectivity determination.

Because there are usually multiple solutions for the linear programming, it would be desirable (1) to enumerate every possible solution and pick a most preferred one, or (2) to modify the linear programming formulation by changing the objective function or adding more constraints that can narrow the solutions to more preferred ones. However, the number of solutions can be large enough to make manual-selection impractical.

Author Contributions: Conceptualization, S.K.J. and Y.K.; methodology, S.K.J. and Y.K.; software, S.K.J.; validation, S.K.J. and Y.K.; writing—original draft preparation, S.K.J.; writing—review and editing, Y.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a grant (20TSRD-B151228-02) from the Urban Declining Area Regenerative Capacity-Enhancing Technology Research Program and by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant (20UMRG-B158194-01) funded by the Ministry of Land, Infrastructure and Transport of the Korean government.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hillier, B.; Hanson, J. *The Social Logic of Space*; Cambridge University Press: Cambridge, UK, 1984.
2. Peponis, J.; Wineman, J.; Bafna, S.; Rashid, M.; Kim, S.H. On the generation of linear representation of spatial configuration. *Environ. Plan. B Plan. Des.* **1998**, *25*, 5595–5676. [CrossRef]
3. Turner, A.; Penn, A.; Hillier, B. An algorithmic definition of the axial map. *Environ. Plan. B Plan. Des.* **2005**, *32*, 4254–4344. [CrossRef]
4. Batty, M.; Rana, S. The automatic definition and generation of axial lines and axial maps. *Environ. Plan. B Plan. Des.* **2004**, *31*, 6156–6240. [CrossRef]
5. Jiang, B.; Liu, X. Automatic generation of the axial lines of urban environments to capture what we perceive. *Int. J. Geogr. Inf. Sci.* **2010**, *24*, 5455–5458. [CrossRef]
6. Liu, X.; Jiang, B. Defining and generating axial lines from street center lines for better understanding of urban morphologies. *Int. J. Geogr. Inf. Sci.* **2012**, *26*, 15211–15232. [CrossRef]
7. Hillier, B.; Penn, A. Dense civilisations: The shape of cities in the 21st century. *Appl. Energy* **1992**, *43*, 416–426. [CrossRef]
8. Penn, A.; Desyllas, J.; Vaughan, L. The space of innovation: Interaction and communication in the work environment. *Environ. Plan. B Plan. Des.* **1999**, *26*, 1932–2018. [CrossRef]
9. Flemming, U. Wall representations of rectangular dissections and their use in automated space allocation. *Environ. Plan. B Plan. Des.* **1978**, *5*, 2152–2232. [CrossRef]
10. Anderssen, R.S.; Ive, J.R. Exploiting structure in linear-programming formulations for land-use planning. *Environ. Plan. B Plan. Des.* **1982**, *9*, 3313–3339. [CrossRef]
11. Anderssen, R.S.; Cocks, K.D.; Ive, J.R. Implications of the generalized upper bounding structure in land-use allocation. *Environ. Plan. B Plan. Des.* **1983**, *10*, 2072–2117. [CrossRef]
12. Erkut, E.; Francis, R.L.; Lowe, T.J. A multi-median problem with inter-distance constraints. *Environ. Plan. B Plan. Des.* **1988**, *15*, 1811–1890. [CrossRef]
13. ReVelle, C.; Snyder, S. A shortest path model for the optimal timing of forest harvest decisions. *Environ. Plan. B Plan. Des.* **1996**, *23*, 1651–1675. [CrossRef]
14. Turner, J.; Wineman, J.; Psarra, S.; Jung, S.K.; Senske, N. *Syntax 2D 1.3.7*; Computer Software; University of Michigan: Ann Arbor, MI, USA, 2006.